System Specification

Constraint Management Phase I Thread, Thor DP1

Checkout and Launch Control System (CLCS)

84K00302-005

Constraint Manager Phase 1 Thread Assessment

March 17, 1998

Version 2.0

Table of Contents

1. Introduction

1.1 Constraint Manager Phase 1 Thread Overview.

The Constraint Manager Phase 1 Thread provides basic Constraint Management application support. Constraint Management provides the capability to monitor one or more FDs for a predetermined condition and notify personnel operating the Test Set, and software applications executing within the Test Set, that the monitored data has transitioned into or out of the constraint condition.

1.2 Constraint Manager Phase 1 Thread Concept

A CLCS constraint is a confinement or restriction placed on data, defined in terms of an expression (boundaries or algorithm) which when violated result in an asynchronous event used for application control or information display. There are two classes of constraint: 1.) constraints which relate to STS operations constraint conditions, 2.) constraints which are used to initiate real-time control events based on FDs in CLCS applications. This thread will focus on the second class of constraint processing for application control.

This thread will provide constraint processing functionality which executes as an independent thread of execution in the DDP during RTPS run-time. Constraint processing shall continuously monitor FD significant change data to detect constraint limit or return to limit violations. It shall receive constraint limit expression specification from system or user applications and provide asynchronous constraint notice back to the specifying or designated application immediately upon violation detection. This thread will also provide for the capability to view instantiated constraints and related historical constraint events.

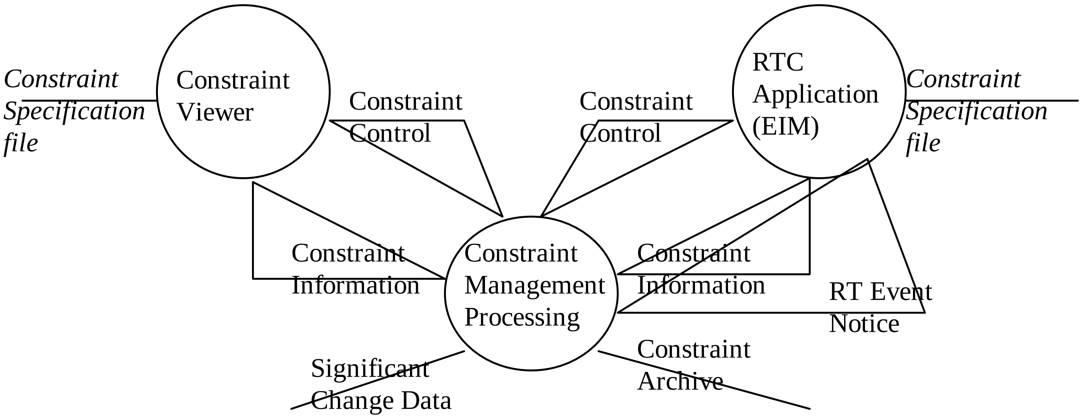

Figure 1-1, Functional Flow Diagram

1.2.1 Interfaces

1.2.1.1 Constraint Control

Constraint control provides the capability for a client application or viewer to assert, release, interrogate, or alter constraint processing. Constraint criteria can be specified at run-time, or can be pre-defined in a constraint file. When the assert request is made, a unique handle (Constraint ID) will be returned.

- Specify (assert) - defines the FD, expression, and attributes associated with constraints. The expression is defined based on limit algorithm, health check, and boundary check. Multiple constraint specifications may be asserted against a single FD.
    - Expression
        - Limit Algorithm - high and low limit, change delta, or logical equivalence.
        - Health Check - test for good or bad health
        - *Boundary Check - optional test for period or frequency violation*
        - Attributes -
        - Viewability - user class
        - RCL -
        - Clients - receiving application
        - Owner - specifying application/viewer
- Release - removes constraint specification from system
- Interrogate - returns constraint specification information
- Alter - provides for modification of the constraint specification and attributes

1.2.1.2 Constraint Information

Constraint information consists of the publication of constraint specification record updates and historical event messages

- Constraint Specification Update Message - latest update message containing the constraint specification record. It is distributed in response to an interrogation or cyclically for viewing.
- Historical Event Message - non-real-time message generated when a transition to or from a constraint violation is detected. It is distributed cyclically for viewing. The message will contain event time, state, count, and attributes.

1.2.1.3 Real-time Event Notice

The real-time event notice is generated when a transition to or from a constraint violation is detected. It is distributed asynchronously to one or more client applications using point to point communications. The real-time event notice will include event time, state, count, and attributes.

- Constraint ID - A unique handle allocated by the system at run time
- Time - System time stamp
- State - Derived from the execution of the constraint expression as an enumerated value
- Value - Value at time of exception
- Count - Number of violations
- Attributes - Viewability, client applications, and owner

1.2.1.4 Significant Change Data

Constraint management processing will reside on the DDP and process all constraints at the system change rate.

1.2.1.5 Constraint Archive (Logging)

All constraint control, constraint information, and RT event notice transactions will be archived for retrieval.

1.2.1.6 Constraint Specification File

Constraint specifications may be predefined in file and invoked from a user application or constraint viewer. An editor will be provided for the Atlas Delivery for users to define a set of constraints. The Editor will perform syntax checks and FD validation.

1.2.2

Constraint Phase 1 Thread Components

1.2.2.1 Test Build

New TCID tables for Constraint Management will be supported:

- *To include constraint files as a user application type associated with an RSYS*
- *A preference table to define a preferred list of Summary Constraint for each User Class, for display purpose.*

1.2.2.2 RTC Applications

Constraint Manager Control

Applications will be able to request for the following Constraint Management support via the use of Constraint Management Services API:

- Assert a new constraint.
- Release an existing constraint.
- Get status of an existing constraint.
- *Test a temporary constraint.*
- Request for definition of a constraint.

1.2.2.3 Constraint Manager Processing at the DDP

Constraint Manager processing at the DDP and CCPs include:

- Process constraint requests from applications.
- Maintain a constraint cache.
- Data Distribution will notify the Constraint Manager when all the FD changed data has been placed in queue for current SSR cycle.
- Monitor FDs specified in the requests.
- Check FD values against the constraint criteria.
- Notify the target application when constraint transition is detected.
- Distribute constraint specification updates **and RT event messages** to all CCP and CCWs across the RTCN and DCN

1.2.2.4 Constraint Manager Processing at the CCWSs

Constraint Manager processing at the CCWSs include:

- Receive change constraints status from the Master CCP across the DCN.
- Maintain a copy of the global constraint cache.

1.2.2.5 Constraint Viewers at the CCWSs

Display support provided by the Constraint Viewer can be subdivided in to two categories called event viewer and constraint control:

- Event Viewer at the CCWSs will display a scrollable history of constraint information via the use of Application Services API, which will in turn call the Constraint Management API to retrieve information from the global constraint cache residing at the local platform. Capabilities accessible from the event viewer include:
  - View most recent constraint event including FD, nomenclature, expression, and value
  - Alter viewability via acknowledge, un-acknowledge, and erase
  - View FD constraint state dynamically including value, count, and series of recent values.
  - Invoke other viewers such as plot and data viewers. Display constraint historical event messages in time order of constraint violation.
  - Displayed constraints will occupy a maximum of two lines and will contain:
    - Constraint descriptor, which is a brief text description of the Constraint
    - RCL indicator
    - Constraint violation/non-violation transition and time of occurrence.
    - Current constraint exception state updated every second.

- Support display of the following from a displayed constraint:
    - Constraint ID
    - Constraint expression
    - Input FDs and their nomenclature
    - Consistent sets of values/health for input FDs and output constraint state including:
        - Time of last transition to violation state
        - Time of last transition out of violation state
        - Current constraint state
- Constraint control can be invoked from constraint viewer. The operators will be able to assert, release, and alter constraints. The requests made at the viewer will be sent to the constraint manager to be monitored.

1.2.2.6 Data Logging at the SDC

The following Constraint related data will be recorded at the SDC:

- Constraint control transactions.
- Real-time event notices
- Time of occurrence of constraint changes.

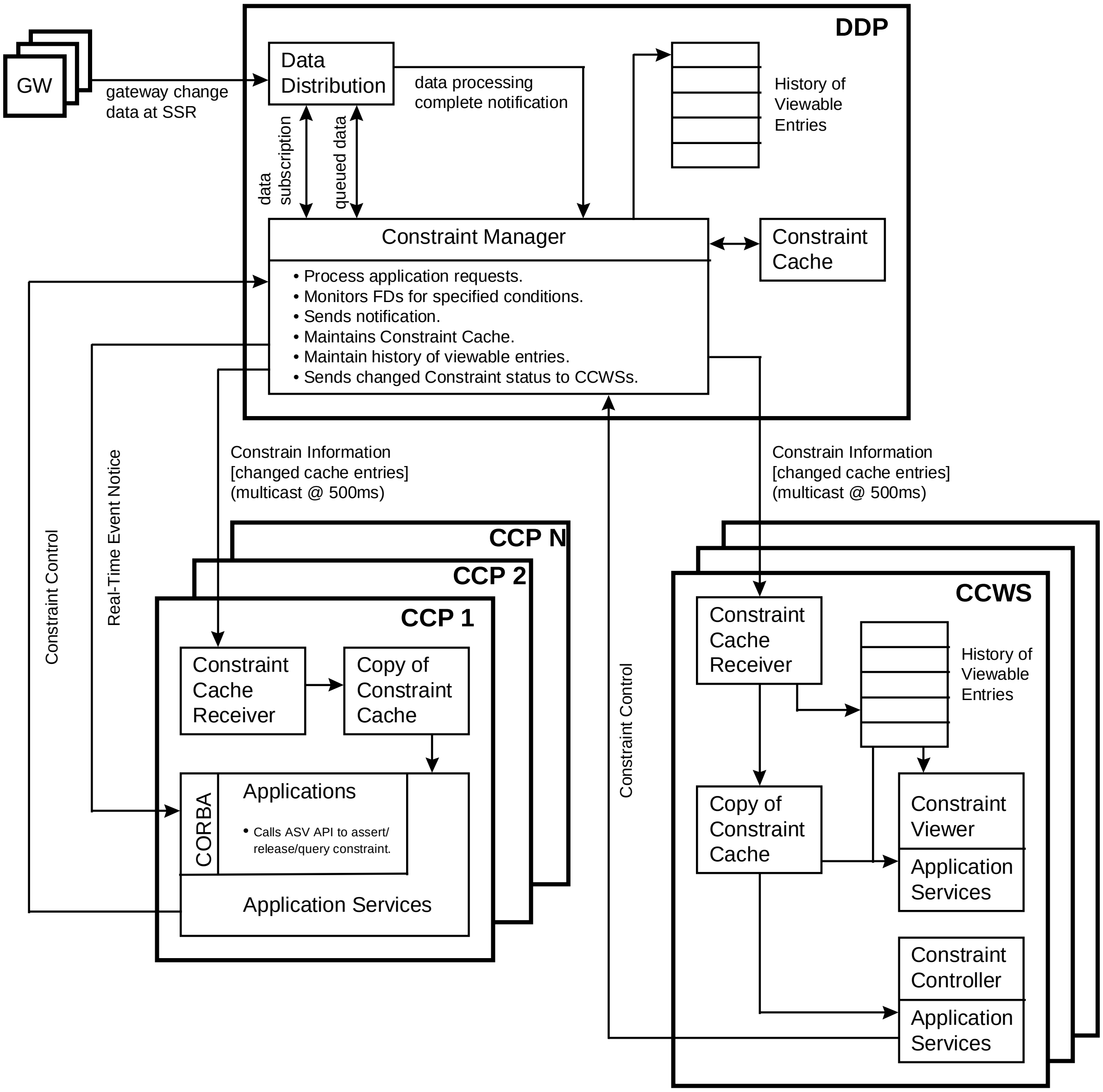
DDP
GW
gateway change data at SSR
Data Distribution
data processing complete notification
data subscription
queued data
History of Viewable Entries
Constraint Manager
• Process application requests.
• Monitors FDs for specified conditions.
• Sends notification.
• Maintains Constraint Cache.
• Maintain history of viewable entries.
• Sends changed Constraint status to CCWSs.
Constraint Cache
Constraint Control
Real-Time Event Notice
Constrain Information [changed cache entries] (multicast @ 500ms)
Constrain Information [changed cache entries] (multicast @ 500ms)
CCP N
CCP 2
CCP 1
Constraint Cache Receiver
Copy of Constraint Cache
CORBA
Applications
• Calls ASV API to assert/ release/query constraint.
Application Services
Constraint Control
CCWS
Constraint Cache Receiver
History of Viewable Entries
Copy of Constraint Cache
Constraint Viewer
Application Services
Constraint Controller
Application Services

Figure 1-2, Constraint Management Interconnect Diagram

1.3

Constraint Manager Phase 1 Thread Specification

1.3.1 Statement Of Work

- Define the list of *logical, mathematical,* and relational function required by the users for Constraint Management.
- Determine if a ControlShell can be utilized and provide the selected tool.
- *Provide the initial Pre-Build Constraint Management Editor.*
- Provide the capability for Fused FDs to be utilized by Constraint Management.
- Provide an initial API for System Viewers with the minimum capability to access Constraint.
- Provide performance data for system modeling.
- Confirm and/or modify system data flow diagrams for Constraint Management and provide the appropriate updates to the SDD.
- Define Packet formats for Constraint Management.
- Provide the capability to monitor Measurement FDs at the rate the data changes and determine when constraint conditions are met.
- Provide the capability for multiple users and system or user applications to request notification of constraint events for each Measurement FD.
- *The CLCS shall provide the capability to monitor measurement data (both converted count data or calibrated engineering units) for out of limits excursions and notify registered uses when any of the following conditions occurs:*
    - *N samples in a row that meet the constraint*
    - *N samples in a given time that meet the constraint*
    - *There is less than a specified time between constraint events*
- Provide the capability to access current constraints and their algorithms.
- Provide the capability to create new constraints from an application, or keyboard.
- Provide logging of error, performance and state change information.
- Baseline system messages using the System Message Catalog to include message and help text.

1.3.2 Requirements

Performance Requirements from SLS

- None assigned for Thor
- Demonstration

Other Constraint Manager Related Requirements from SLS

2.2.4.3.8 The RTPS shall provide a Constraint Monitor Viewer which provides a mechanism for asserting and viewing constraints against measurement FDs for Constraint Monitor purposes only.

2.2.5.4.1 The RTPS shall provide the capability to monitor Measurement FDs at the rate the data changes and determine when predefined constraint limits are exceeded or constraint conditions are met.

2.2.5.4.1 The RTPS shall provide the capability for multiple (TBD number) users and system or user applications to request notification of constraint events for each Measurement FD.

2.2.5.4.1 The RTPS shall allow an application to set more than one set of limits on any Measurement FD.

2.2.5.4.2 The RTPS shall provide the capability for each user, and system or user application requesting constraint notification to specify the limits/condition under which they will be notified.

2.2..5.4.2 The RTPS shall provide the capability to monitor measurement data (both converted count data or calibrated engineering units) for out of limits excursions and notify registered users when any of the following conditions occur:

1. *N samples in a row that meet the constraint*
2. *N samples in a given time that meet the constraint*
3. *There is less than a specified time between constraint events*

2.2.5.4.3 CLCS shall provide the user the capability for any user and system or user application to create new constraints from an application, keyboard, *or predefined file*, determine and/or view the current constraints and their algorithms, modify the list of constraints, and select the algorithms relating to them.

2.2.5.4.4 The RTPS Constraint Management function shall be fault tolerant.

2.2.5.4.5 *The RTPS shall provide the capability to test for or view constraint violations at a summary level (e.g. Launch Commit Criteria or OMRSD).*

2.2..5.7.4 RTPS shall utilize Constraint Management for constraint monitoring for a Test Application Script.

1.4 Constraint Manager Phase 1 Thread Hardware Diagram

Not Applicable.

1.5

Constraint Manager Phase 1 Thread Deliverables

Deliverable	R&D Document	Code	API Manual	Users Guide
Data Distribution & Processing - Constraint Manager CSC	X	X	X	N/A
Data Distribution & Processing - Data Distribution CSC	X	X	X	N/A
Application Services ■ Application Services ■ EIM Services ■ User Display Services	X	X	X	N/A
Constraint Viewer	X	X	N/A	X

1.6 Constraint Manager Phase 1 Thread Assessment Summary

This section contains the summary of the costs and labor involved in implementing the Data Distribution Completion Thread capabilities. It is broken into three sections. The first is a summary of the individual CI (CSCI and HWCI) labor assessments. The second is a summary of hardware costs. The third is a summary of procurement activities needed.

1.6.1 Labor Assessments

The total Labor Costs required to provide this capability are summarized in the following table:

No.	CSCI/HWCI Name	Thor LM	Changes covered in
1	Data Distribution & Processing CSCI - Constraint Manager CSC	24	
2	Data Distribution & Processing CSCI - Data Distribution CSC	1	
3	Application Services CSCI - FD Services	2	
4	Application Services CSCI - User Display Services CSC	11	
5	Application Services CSCI - Constraint Management Services CSC	5.5	
6	Application Services CSCI - EIM Services CSC	0	EIM and Application Interface Thread
7	Constraint Viewer	6	
	TOTAL	49.5	

1.6.2 Hardware Costs

None.

1.6.3 Constraint Manager Phase 1 Thread Procurement

None.

1.7

Constraint Manager Phase 1 Thread Schedule & Dependencies

1.7.1 Schedule

Task Name	Start	Finish
Thor Assessment Kickoff	07/23/97	07/23/97
Concept Panel Internal Review	10/22/97	10/22/97
Concept Panel	10/24/97	10/24/97
Thor Development		
Requirement Panel Internal Review	11/12/97	11/12/97
Requirement Panel	11/14/97	11/14/97
Design Panel Internal Review	11/24/97	11/24/97
Design Panel	11/26/97	11/26/97
CSCI Unit Testing	01/12/98	01/30/98
CSCI Development Integration Test (UIT)	02/02/98	02/14/98
CSCI Formal Integration Test (CIT)	02/16/98	02/20/98
Support System Integration Test	02/23/98	03/27/98
Thor Development Complete	03/27/98	03/27/98

1.7.2 Dependencies

The following dependencies are essential to the UIT and CIT schedule listed on the above section:

No.	Dependency Area	Dependency	Need Date
1		•	
2		•	01/02/97
3	User Applications	Needed for UIT • Example C++ & C Applications, EIM applications and User Display applications calling Application Services API to make Constraint requests and receive Constraint notifications.	02/02/98
4	• Application Services • User Display Services • EIM Services	Needed for UIT • APIs to make Constraint requests. • APIs to receive Constraint notifications.	02/02/98
5	System Services	Needed for later part of Unit Test • CORBA compliant interface to send Constraint notifications.	01/23/98
6		•	
7	Constraint Viewer	Needed for UIT.	02/02/98
8	Data Distribution CSC	Needed for Unit Testing • Test Tool to drive various FD values to the Constraint Manager. • Notify the Constraint Manager when data is ready for CM processing.	01/02/98
9	SDC	Needed for CIT testing • Capability to record Constraint requests, changes in Constraint states, and time of occurrences of Constraint changes.	02/02/98

No.	Dependency Area	Dependency	Need Date

1.8 Constraint Manager Phase 1 Thread Simulation Requirements

The following example applications will be used to simulate application testing.

- Example C++/C common applications
- Example EIM applications
- Example User Display applications

The Data Distribution Test Tool will be used to drive various FD values to the Constraint Manager for a representative set of FDs to transition into and out of constraint violation states.

1.9 Constraint Manager Phase 1 Thread Integration and System Test

Constraint Manager end-to-end development testing will be performed prior to start of System Test:

- Example applications will be used to issue Constraint Assert/Specification request to the Constraint Manager at the DDP, via the user of Application Services APIs.
- Data Distribution Test Tool will be used to drive various data values to a representative set of FDs to test various limit processing.
- Snap shot of the Constraint Cache will be taken at different phases of the test to ensure all Constraint information are current in the cache.
- Snap shot of the copy of Constraint Cache at the CCWSs will be taken at different phases of the test to ensure the copy is identical to the Cache copy at the DDP with exception of request made within the last 500 ms.
- The Constraint Viewer will be used to compare the displayed constraint information against those in the Constraint Cache.

System test plan and test procedures will be prepared by the System Test Organization.

1.10 Constraint Manager Phase 1 Thread Training Requirements

None.

1.10.1 Training Needed

None.

1.10.2 Training to be provided

- Training on using the Constraint Management and Applications Services APIs can be provided upon request.
- Training on using the Constraint Viewer will be provided upon request.

1.11 Constraint Manager Phase 1 Thread Facilities Requirements

Additional/replacement resources will be needed by the Constraint Manager Phase 1 Thread for SDE-JSC facility:

- Needed resources are covered by the Data Distribution Completion Thread.

1.12 Travel Requirements

This section contains a list of travel requirements.

From	To	Reason	No. People	Duration	Est. Date or Frequency
Houston	KSC	Support of Design Panels by Houston developers/management	2	3 days per trip	3 trips
Houston	KSC	On-site integration testing and system testing by Houston developers	3	3 weeks per trip	2 trips
Houston	KSC	On-site training for developers/management	1	1 week	1 trip

1.13 Constraint Manager Phase 1 Thread Action Items/Resolution

1. Clarification/definition of how user class and RSYS will be used in CLCS (display, authentication, etc.) is needed.

2. CSCI Assessments

2.1 Data Distribution & Processing CSCI Assessment

The following capabilities will be provided by the Data Distribution CSCI in support of the Data Distribution Thread:

Data Distribution CSC Work Required

The Data Distribution CSC will be enhanced to support the following:

- Notify the Constraint Manager when data is ready for CM processing.
- Test Tool to drive various FD values and health status to the Constraint Manager for testing purpose.

Constraint Manager CSC Work Required

The Constraint Manager CSC will provide the following:

- Constraint Manager processing at the DDP which include:
  - Process constraint specifications from the RT applications and the Viewer.
  - Maintain a constraint cache.
  - Monitor FDs.
  - Check FD values against the constraint criteria.
  - Notify the target application when constraint transition is detected.
  - Distribute constraint specification updates and RT event meesages to all CCP and CCWs across the RTCN and DCN

- Constraint Manager processing at the CCPs
  - Receiver changed constraint specification updates from the RTCN
  - Maintain a copy of the Constraint cache

- Constraint Manager processing at the CCWS
  - Receive changed constraint specification updates from the DCN
  - Maintain a copy of the global constraint cache.

- Programmatic interface that allows the applications to:
  - Send a request to the Constraint Manager to assert/release/interrogate a constraint (refer to the ‘RTC Application’ section and the ‘Constraint Information’ section of the Concept document for details).

CSCI Assessment

Data Distribution CSC	CSC Labor (LM)	% of CSC
Data Distribution CSC ■ Test Tool to drive data to CM ■ Notification to CM	1 LM	
TOTAL	**1 LM**	

Constraint Manager CSC	CSC Labor (LM)	% of CSC
Constraint Manager CSC ■ CM processing at the CCP and DDP ■ CM processing at the CCP ■ CM processing at the CCWSs ■ API	24 LM	
TOTAL	**24 LM**	

Basis of estimate

Estimate is based on a combination of the following:

- Source lines of code
- Complexity factor
- Amount of integration with other CSCs

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	50
Users Guide	N/A	N/A
API Interface Document	New	10
Interface Design Document	N/A	N/A
Test Procedure	New	30

Assumptions

Refer to the Constraint Manager Phase 1 Thread Action Items and Resolution Section (Section 1.13).

Open Issues

Refer to the Constraint Manager Phase 1 Thread Action Items/Resolution items in Section 1.13.

2.2 System Viewer CSCI Assessment

Constraint Viewer Work Required

Display support provided by the Constraint Viewer can be subdivided in to two categories called event viewer and constraint control:

- Event Viewer at the CCWSs will display a scrollable history of constraint information via the use of Application Services API, which will in turn call the Constraint Management API to retrieve information from the global constraint cache residing at the local platform. Capabilities accessible from the event viewer include:

- View most recent constraint event including FD, nomenclature, expression, and value
- Alter viewability via acknowledge, un-acknowledge, and erase
- View FD constraint state dynamically including value, count, and series of recent values.
- Invoke other viewers such as plot and data viewers. Display constraint historical event messages in time order of constraint violation.
- Displayed constraints will occupy a maximum of two lines and will contain:
  - Constraint descriptor, which is a brief text description of the Constraint
  - RCL indicator
  - Constraint violation/non-violation transition and time of occurrence.
  - Current constraint exception state updated every second.
- Support display of the following from a displayed constraint:
  - Constraint ID
  - Constraint expression
  - Input FDs and their nomenclature
  - Consistent sets of values/health for input FDs and output constraint state including:
    - Time of last transition to violation state
    - Time of last transition out of violation state
    - Current constraint state

- Constraint control can be invoked from constraint viewer. The operators will be able to assert, release, and alter constraints. The requests made at the viewer will be sent to the constraint manager to be monitored.

This is a list of work to be accomplished for this function.

System Viewers Assessment

CSC Name	CSC Labor (LM)	% of CSC
Constraint Viewer	6 LM	

Basis of estimate

Code will be generated by use of a GUI design tool making a code estimate impractical.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	TBD
Users Guide	New	TBD
API Interface Document	N/A	N/A
Interface Design Document	New	TBD
Test Procedure	New	TBD

Assumptions

None

Open Issues

None

2.3 Application Services CSCI Assessment - To Be Provided

Application Services Work Required
Applications services provides the layer that interfaces the functionality of System Services, System Applications, and User Applications. For Constraint Management support, ASV will provide support hooks for Control Shell (or other selected tool) to access CLCS objects. ASV will also provide the mechanisms for applications to easily and safely access constraints and the constraint manager.

FD Services (FDS) Work Required:

- (TBD) provide a service to do "authentication lite". This API will give an indicator that the RSYS associated with a given FD is a member of the set of RSYSs associated with the user class of the user.

User Display Services (UDS) Work Required:

User Display Services must provide support for the Data Fusion Viewer which is written in Java. The UDS work includes:

- providing a JAVA data handler, and any other JAVA-unique items needed
- provide APIs to give constraint constraint-related data (historical and current) to viewers (details TBD)

Constraint Management Services (CMS) Work Required:

Constraint Management Services provides constraint-unique support including:

- APIs to construct or destroy a constraint
- APIs to query the status or state of a constraint
- API to alter a constraint

End Item Manager Services (EIS) Work Required:

End Item Manager Services is tasked to provide the Control Shell constraint event "tossers" and "catchers". These are the mechanism by which Control shell accesses FDs. This work has been included under the EIM and Application Interface Thread.

Applications Services Assessment

CSC Name	CSC Labor	% of CSC
FDS	2	
UDS	11	
CMS	5.5	
EIS	0	
ASV Total	18.5	

Basis of estimate

Estimate is based on Redstone experience.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New and update	40
Users Guide	N/A	
API Interface Document	New and update	50

Document Type	New/Update	Number of Pages
Interface Design Document	New and updates	TBD
Test Procedure	New test cases	100

Other:

Assumptions

- Estimate dependent upon use of Control Shell.
- No impacts due to delivery (or non-delivery) of Control Shell updates/upgrades.

Open Issues

- Should “authentication lite” be in the system?

Assessment Provided By: Julia Samson, ASV CSCI Lead, 1-2212

2.4 SDC Assessment

SDC Work Required

- Record Constraint specification updates, changes in Constraint state, and time of occurrence for Constraint changes.

Per Earl Foster, the Constraint Specification updates will automatically get recorded as long as the Packet Payload format is followed. In this case, the assessment is 0 for Constraint Manager.

3. HWCI Assessments

Not Applicable.